

# Automatinio sprendimų vertinimo informatikos olimpiadose raida ir perspektyvos

## Jūratė Skūpienė

Matematikos ir informatikos instituto jaunesnioji mokslo darbuotoja  
Institute of Mathematics and Informatics, Younger research fellow  
Akademijos g. 4, LT-08663, Vilnius  
Tel. (+370 5) 210 97 32  
El. paštas: jurate.sk@gmail.com

*Pastaruoju metu mokslinėje visuomenėje vis daugiau atgarsio sulaukia įvairios informatikos (algoritavimo) olimpiados. Ši sritis tampa mokslinių tyrimų objektu. Informatikos olimpiadų dalyviai turi sukurti algoritmus ir juos užrašyti nepriekaištingai veikiančiomis programomis viena iš leidžiamų programavimo kalbų. Darbai yra vertinami automatiškai testuojant juodosios dėžės principu, visiškai nebandant analizuoti paties algoritmo. Toks vertinimas sulaukia nemažai kritikos, tačiau kol kas gerų alternatyvų nėra. Straipsnyje pateikiama automatinio vertinimo informatikos olimpiadose raida nuo pirmųjų olimpiadų iki šių dienų, apžvelgiami darbai, kuriuose nagrinėjama ši problematika, aptariamos olimpiadose ir mokslinėje literatūroje siūlytos patobulintos vertinimo schemos, numatomos ateities gairės.*

Pasaulinės informatikos olimpiados (*International Olympiad in Informatics* – IOI) yra vienas labiausiai prestižinių mokiniams skirtų kompiuterių mokslo (algoritmų kūrimo) konkursų (*International Olympiad...*, 2007). Šią olimpiadą 1989 metais inicijavo UNESCO. IOI rengiama kartą per metus vis kitoje pasaulio šalyje, į ją suvažiuoja dalyviai iš septyniasdešimties–aštuoniasdešimties pasaulio šalių. Olimpiadoje varžomasi individualiai dvi dienas. Varžybos trunka penkias valandas ir per jas dalyviams tenka spręsti tris uždavinius, iš viso šešis uždavinius per dvi dienas. Visi pateikiami uždaviniai yra algoritminio pobūdžio. Dalyvis turi sukurti pateiktos uždotos algoritmą ir jį užrašyti viena iš leidžiamų programavimo kalbų (Paskalio, C arba C++).

Dalyvių algoritmai, užrašyti programomis, IOI vertinami automatiškai taikant *juodosios dėžės* principą. Kiekvienam uždaviniui iš anksto parengiami pradiniai duomenų rinkiniai (testai). Kiekviena programa sukompiliuojama ir įvykdoma su kiekvienu testu. Jei gaunamas teisingas

rezultatas, už tą testą skiriami taškai, jei ne – balų neskiriama.

Remiantis IOI modeliu organizuojama daug regioninių (pvz., Baltijos, Vidurio Europos, Balkanų, Ramiojo vandenyno valstybių, Afrikos–Arabų ir kt.) bei nacionalinių olimpiadų (Honk Kong..., 2007). Taip pat vyksta ir įvairių neakivaizdinių kompiuterių mokslo konkursų mokiniams, yra portalų, kuriuose remiamasi panašiais principais kaip ir IOI (pvz., USACO, UVA ES, ICPC). Lietuvos moksleivių informatikos olimpiada taip pat rengiama remiantis šiuo modeliu, tik kiek pakeistu (pvz., yra dvi amžiaus grupės, šiek tiek kitoks vertinimas ir pan.) (*Lietuvos moksleivių...*, 2007). Todėl toliau, analizuodami programavimo olimpiadų ir varžybų specifiką, jų problematiką, aptarsime IOI patirtį turėdami omeny, kad tos pačios problemos būdingos ir daugeliui kitų panašaus pobūdžio varžybų ir olimpiadų.

Pastaruoju metu automatinis darbų vertinimas ir jo patikimumas kelia vis daugiau klausimų

ir diskusijų (Verhoeff T., 2006). Nuo 2006 m. sausio, kai buvo surengtas pirmasis tarptautinis mokslinis seminaras pasaulinių olimpiadų vertinimo ir kitoms problemoms aptarti, IOI dalyvaujantys mokslininkai informatikos olimpiadas pradėjo traktuoti, kaip mokslinio tyrimo objektą. Nuo tada konferencijose nuolat analizuojamos informatikos olimpiados įvairiais moksliniais aspektais, žurnaluose pasirodo mokslinių straipsnių.

### **Automatinio vertinimo informatikos olimpiadose raida. Pirmasis etapas (iki PVAS – Programavimo varžybų sistemų atsiradimo), 1989–2000 metai**

Olimpiadose dalyviai turi sukurti algoritmus ir juos užrašyti veikiančiomis programomis. Tad svarbiausi tikslai vertinant dalyvio darbą yra tokie: nustatyti, koks algoritmas yra realizuotas programoje, įvertinti jo tinkamumą nagrinėjamam uždaviniui spręsti. Algoritmo tinkamumas nusakomas dviem aspektais – algoritmo teisingumu (ar bet kuriems pradiniais duomenims, tenkinantiems sąlygoje patektus ribojimus, algoritmas pateikia teisingą rezultatą) ir efektyvumu (kokio sudėtingumo yra šis algoritmas). Reikėtų atkreipti dėmesį: olimpiadose pateikiami tokie uždaviniai, kad mokiniai sukonstruotų visiškai teisingą jų sprendimo algoritmą. Uždavinių, kuriems neegzistuoja teisingas ir efektyvus algoritmas (t. y. kuriems išspręsti būtinai reikia euristinių algoritmų), vengiama. Kadangi algoritmas užrašytas programa, vertinant turėtų būti atkreipiamas dėmesys ir į algoritmo realizacijos ypatumus: bendrumą, aiškumą, paprastumą.

Pirmosiose Lietuvos informatikos olimpiadose dalyviai programas rašydavo ant popieriaus lapo ir, suprantama, kiekvieną algoritmą (programą) analizuodavo vertinimo komisija. Toks vertinimas turėjo nemažai pranašumų: kiekvienas algoritmas būdavo gerai išanalizuojamas, buvo galima išskirti teisingus, iš dalies teisingus, klaidingus algoritmus. Nesunkiai būdavo įvertinami įvairūs realizacijos aspektai, programavimo stilius. Vertinant nekreipta dėmesio į smulkias

realizavimo klaidas. Prastai ir nesuprantamai realizuoti algoritmai būdavo atmetami.

Tiesa, toks vertinimas turėjo ir trūkumų. Algoritmų teisingumo įrodymai, kurie pateikiami moksliniuose straipsniuose ar vadovėliuose aukštųjų mokyklų studentams, yra per sudėtingi vidurinių mokyklų ir gimnazijų studentams. Tad kartu su sprendimais jie turėjo pateikti trumpą sprendimo aprašymą ir pačią programą, tačiau pagrindimo nereikėjo. Dalyviai sukonstruodavo algoritmų, kurių teisingumas ar klaidingumas nebuvo akivaizdus. Norint tai patikrinti, būtų reikėję remtis moksliniais metodais. Tokiais atvejais būdavo neaišku, kaip įvertinti algoritmą. Olimpiadai plečiantis ir sunkėjant uždaviniams, per trumpą laiką kokybiškai išanalizuoti daug programų tapo vargiai įmanoma. Nors labai prastai realizuoti sprendimai būdavo atmetami ir nevertinami, tačiau tik nedidelė dalis programų būdavo parašytos puikiu programavimo stiliumi. Vidutinišku stiliumi užrašytų sprendimų būdavo nemažai, o jų analizė užimdavo gana daug laiko.

Toks vertinimas dar mažiau įmanomas tarptautinėse olimpiadose dėl daugiakultūrės aplinkos. Iš skirtingų valstybių atvykę dalyviai identifikatorių vardais gali parinkti savo gimtosios kalbos žodžių ar jų atitikmenų, jie atstovauja skirtingiems programavimo stiliumis, kas dar labiau sunkintų vertinimą. Pirmosiose dviejose pasaulinėse informatikos olimpiadose (1989–1990 m.) dalyviai turėjo pateikti ne tik programas, bet ir žodinius algoritmų aprašymus. Į anglų kalbą žodinius aprašymus versdavo komandų vadovai ir vertinimo metu būdavo galima diskutuoti su vertintojais dėl aprašomo algoritmo, jo efektyvumo. Tačiau tolesnėse IOI pastebėjus, kad vertinimas labai priklauso ne tik nuo paties aprašo, bet ir nuo dalyvių bei komandų vadovų rašytinės ir šnekamosios anglų kalbos žinių, o be to, tai neišsprendė klausimo, kaip vertinti idėjas, kurių teisingumas/klaidingumas nėra akivaizdus, to buvo atsakyta. Lietuvos informatikos olimpiadose iki šiol išliko ir tebėra labai svarbi vertinimo dalis.

Nuo trečiosios IOI (1991 m.) visi dalyvių pateikti sprendimai vertinami tik juodosios dėžės

principu, t. y. testuojant. Pats programos tekstas nėra analizuojamas ir peržiūrimas. Testavimas būdavo vykdomas beveik „rankiniu“ būdu. Dalyvio akivaizdoje būdavo sukompiliuojama jo programa ir vykdoma su kiekvienu iš testų. Dalyvis matydavo, jeigu jo programa nekorektiškai baigdavo darbą ar iš viso jo nebaigdavo. Dalyvis galėdavo iš karto pasitikrinti, kodėl jo programa pateikdavo vienokį ar kitokį rezultatą. Pačiose pirmosiose olimpiadose buvo stengiamasi atskirti matematinę algoritmo efektyvumą sąvoką nuo jo realizacijos efektyvumo. Laikytasi nuostatos, kad jei algoritmas neefektyvus (pvz., eksponentinio sudėtingumo), tai kad ir kiek ilgai jį vykdysi, jis vis vien nebaigs darbo su didesniais testais per priimtina laiką. O jei algoritmas efektyvus – tai realizacijos detalės, galinčios turėti įtakos jo efektyvumui, nėra svarbios. Tad programos vykdymo laikas nebūdavo nurodomas ir programos vykdomos, kol užtekėdavo laiko, pavyzdžiui, per visą naktį. Vėliau, didėjant kompiuterių procesorių pajėgumams, neribojant laiko tapo sunku atskirti, pavyzdžiui, kvadratinio ir kubinio sudėtingumo sprendimus ir kiekvienam uždaviniui buvo nustatytas fiksuotas vykdymo laikas (pvz., 2 sekundės), kurį viršijusi programa nutraukiama ir už tą testą balų neskiriama.

Tik 1999 metais buvo sukurtos pirmosios automatinės sprendimų testavimo programos, kurios per tinklą pasiekdavo visų dalyvių darbus, juos kompiliuodavo ir lygiagrečiai vykdavo su visais testais. Tai nepakeitė vertinimo esmės, tačiau sutaupydavo laiką ir dalyviai, užuot stovėję už durų ir laukę savo testavimo eilės, iš karto gaudavo lapą su testavimo rezultatais.

Perėjus prie išskirtinai automatinio programų vertinimo, įvairios duomenų formato detalės, kurios iš esmės nėra svarbios realizuojamo algoritmo požiūriu, ėmė darytis labai svarbios vertinant. Pavyzdžiui, neteisingai užrašius duomenų failo vardą arba bandant skaityti duomenis iš dviejų duomenų failo eilučių (kai sąlygoje nurodyta, kad pradiniai duomenys bus įrašyti tik pirmoje failo eilutėje), programa neišveikdavo nė vieno testo ir būdavo įvertinama nulių balų.

Perėjus prie algoritmų vertinimo juodosios dėžės principu, kai kurie vertinimo kriterijai (algoritmo identifikavimas, realizacijos bendrumas, aiškumas, paprastumas) buvo pamiršti. Ką tik minėtų kriterijų nebebūdavo paisoma. Dalyviams ir organizatoriams labiausiai rūpėjo akivaizdžiai neobjektyvus vertinimas, kai dėl korektūros klaidų būdavo prarandami visi taškai. Dalyviai rašė apeliacijas, o mokslinis komitetas svarstė, ką laikyti korektūros klaida (pvz., ar laikyti korektūros klaida, jei duomenų failo varde pakeista viena raidė, jei duomenys skaitomi ne iš vienos, o iš kelių eilučių, jei reikia išvesti didėjimo tvarka surikiuotą seką, o seka pateikta mažėjimo tvarka, jei atsitiktinai sukeisti ciklą kintamieji, jei aprašytas *integer* tipo duomenis, o reikia naudoti *longint* tipo duomenis ir t. t.). Jei sprendimus analizuotų žmogus, daugelyje tokių situacijų šios klaidos būtų priskirtos korektūros klaidoms ir už jas balai būtų sumažinami labai nedaug. Deja, to nevyko IOI. Dalyviai susidurdavo su dar viena nemalonia situacija. Padarę nedidelę loginę klaidą (pvz., masyvas aprašytas nuo pirmojo elemento, o bandomas inicializuoti nulinis elementas), kuri dažnai pastebima tik vykdant programą kitame kompiuteryje (t. y. ne tame pačiame kompiuteryje, kuriame programa buvo kuriama ir derinama), jie taip pat prarasdavo visus taškus.

Norėdami pertestuoti darbą su korektūros klaida, vertintojai turėdavo keisti programos tekstą ir vis sunkiau tapo nustatyti ribą, kur yra korektūros klaida, o kur ne. IOI mokslinis komitetas būdavo linkęs visiškai nieko nekeisti dalyvių programose. Problemos sprendimą pirmieji pasiūlė turkai 1999 metais Beleke vykusioje IOI. Kiekvienam uždaviniui jie pabandė parašyti po duomenų bei rezultatų formatą tikrinančią programą, ir dalyvis gaudavo tokių programų paketą. Su savo programa įvykdęs pavyzdinį testą, galėdavo paleisti rezultato formatą tikrinančią programą ir pasižiūrėti, ar jo programa išveda rezultatą tinkamu formatu. Deja, šios programos negalėjo patikrinti nei formato klaidų skaitant duomenis, nei minėtų loginių klaidų. Tačiau tai buvo pirmasis žingsnis kuriant *Programavimo varžybų aptarnavimo sistemas*.

## **Automatinio vertinimo informatikos olimpiadose raida. Antrasis etapas (Programavimo varžybų sistemų atsiradimas) 2001–2005 metai**

Pirmąją programavimo varžybų aptarnavimo sistemą (PVAS, angl. *Contest and Grading System*) 2001 metais sukūrė R. Kolstad ir ji pirmą kartą buvo panaudota 2001 metais Suomijoje vykusioje IOI. PVAS – tai serverio aplikacijų ir modulių rinkinys, skirtas užtikrinti olimpiadų (varžybų) metu reikalingoms funkcijoms. Prie PVAS prisijungama per tinklą, kiekvienas dalyvis prisijungia iš savo kompiuterio. Norėdamas atiduoti savo sprendimą, dalyvis nurodo failą, kuriame yra jo parašyta programa, ir ją išsiunčia į varžybų serverį. Gauta programa serveryje sukompiliuojama ir įvykdoma su sąlygoje pateiktu pavyzdiniu testu (testais). Jei dalyvio programos negalima sukompiliuoti ar vykdydama pavyzdinį testą programa strigo (lūžo) arba pateikė klaidingą rezultatą, sistema apie tai praneša ir programos nepriima. Dalyvis turėtų bandyti pataisyti savo programą (suprantama, kol nesibaigė varžybų laikas) ir vėl perduoti sprendimą sistemai. To paties uždavinio sprendimą moksleivis gali perduoti sistemai daug kartų, tai neturi įtakos rezultatams. Vertinamas tik paskutinis priimtas sprendimas.

Pradėjus naudoti šią sistemą atkrito daug problemų. Iki minimumo sumažėjo neobjektyvus vertinimo, susijusio su korektūros klaidomis. Sakykime, užrašius klaidingą failo vardą, sistema iš karto praneša apie klaidą ir dalyvis gali ją tuoj pat ištaisyti. Lengvai ištaisomos ir įvedimo/išvedimo formato klaidos. Negana to, dalyvio programa vykdoma kitame kompiuteryje (vertinimo mašinoje), dėl to dažnai galima pastebėti ir smulkias logines klaidas. Pavyzdžiui, tokia klaida kaip neinicializuoto kintamojo reikšmės naudojimas kaip tik lengviau pastebima vykstant programą ne savo kompiuteryje. Sistemos naudojimas leido dalyviams lengvai išspręsti visas su duomenų/rezultatų formatu susijusias problemas ir susitelkti į algoritmo kūrimą.

„Tokia sistema atvėrė naują istorijos puslapį pasaulinių olimpiadų istorijoje“ (Piele, 2001)

ir ne tik išsprendė daugelį vertinimo problemų, bet ir leido pereiti prie darbo „Linux“ aplinkoje. Sukūrus pirmąją sistemą ir pastebėjus jos pranašumus, buvo imta kurti analogiškas sistemas vis jas papildant įvairiomis reikalingomis savybėmis. Pavyzdžiui, galimybė dalyviams pateikti klausimus per sistemą arba analizavimo režimą – kai pasibaigus olimpiadai dalyvis gali pasiekti visas savo pateiktas programos versijas ir sužinoti, kiek taškų jis būtų gavęs už kiekvieną konkrečią versiją. Analogiškas sistemas įvairioms olimpiadoms ir varžyboms sukūrė korėjiečiai (IOI 2002 Contest...), keletą sistemų įvairioms operacinėms sistemoms – lenkai (SIO.NET project., 2007), specialiai Baltijos šalių olimpiadoms sistemas sukūrė Vokietija, Latvija, Estija. Daugelis valstybių, iš jų ir Lietuva, nacionalinėms olimpiadoms organizuoti sukūrė arba ėmė naudoti kitų parašytas analogiškas varžybų sistemas (Skūpienė, 2004).

## **Automatinio vertinimo informatikos olimpiadose raida. Trečiasis etapas (Objektyvių vertinimo schemų paieška), 2006 – iki dabar**

Atsiradusios PVAS iš esmės išsprendė vertinimo problemas, susijusias su duomenų/rezultatų formatu, ir sudarė galimybes dalyviams sutelkti dėmesį į kuriamą algoritmą, taip pat leido ir vertintojams grįžti prie tikrųjų vertinimo tikslų: nustatyti, koks algoritmas yra realizuotas programoje, ir įvertinti jo tinkamumą nagrinėjamam uždaviniui spręsti. Apie tai buvo pradėta kalbėti iš karto atsiradus pirmosioms PVAS.

Automatinio testavimo problema yra ta, kad jis negali garantuoti tinkamo programos teisingumo ir efektyvumo įvertinimo. Be to, kiti programos aspektai – programos aiškumas, bendrumas, paprastumas – taip pat nėra vertinami. Siekiant kuo objektyvesnio vertinimo, imtos kurti ir analizuoti įvairios vertinimo schemas, nagrinėjami konkretūs uždaviniai. Pasirodė pirmieji moksliniai straipsniai ir tyrimai šia tema (Horvath, Verhoeff, 2002; Horvath, Verhoeff, 2003). 2006 metais Vokietijoje buvo surengtas pirmasis informatikos olimpiadų problematikai skirtas tarptautinis seminaras, kuriame buvo aiškiai pasakyta,

kad olimpiadų vertinimo problematika iš esmės turėtų būti suprantama kaip mokslinio tyrimo objektas ir jai turi būti skiriama pakankamai dėmesio. Tai atvėrė kelią tolesniems tyrimams ir moksliniams straipsniams šia tema (Cormak, 2006; Dagienė, Skūpienė, 2004; Opmanis, 2006 ir t. t.).

Vertinant sprendimus tik automatiniu būdu, buvo performuluoti vertinimo tikslai, nes dalies jų vertinant tik juodosios dėžės principu įgyvendinti negalima (kaip ir nustatyti, koks tiksliai algoritmas realizuotas, arba įvertinti programavimo stilių). Tad buvo suformuluoti paprastesni tikslai, kuriuos buvo tikimasi įgyvendinti, kad padėtų tinkamai surinkti dalyvius ir išrinkti nugalėtoją: a) atskirti teisingus algoritmus nuo klaidingų, t. y. parašęs teisingą ir bent nuosaikiai efektyvų sprendimą gauna daugiau taškų už parašiusį klaidingą sprendimą; b) už programą surinktas taškų skaičius turi būti proporcingas asimptotiniam algoritmo efektyvumui; c) už smulkias klaidas (pvz., praleistas ribinis atvejis) dalyvis netenka tik keleto taškų. Manoma, kad šių tikslų sėkmingiausiai galima pasiekti kokybiškai parenkant testus (pradinius duomenis). Gerai parinkus testus, klaidinga programa turėtų bent dalies jų neišveikti. Parinkus skirtingo dydžio (sudėtingumo) testus, turėtų būti galimybė įvertinti efektyvumą. Deja, tai tik teoriniai tikslai. Olimpiadose dažnai nutinka taip, kad klaidinga programa surenka daug taškų, o teisingas algoritmas, turintis nedidelę realizacijos klaidą, įvertinamas mažu balų skaičiumi (Verhoeff, 2006; Forišek, 2006). Siekiant kuo didesnio objektyvumo, IOI buvo naudojamos įvairios vertinimo schemos.

*10–20% taisyklė.* Remiantis šia schema teisingos bet neefektyvios programos turėtų surinkti ne daugiau kaip 10–20% balų. Schema realizuojama parenkant pradinius duomenis taip, kad didžioji dalis testų tikrintų efektyvumą. Šios schemos buvo atsisakyta 2003 metais motyvuojant tuo, kad uždaviniai daugumai dalyvių yra sunkūs\*, o taškų skaičius, skiriamas už teisingas,

bet nepakankamai efektyvias programas, yra per mažas. Pirmąjį argumentą galima ginčyti, tuomet reikia keisti uždavinių atrankos principus, o ne vertinimo schemą. Tačiau su antrąja pastaba tenka sutikti. Kreipiant ypatingą dėmesį į algoritmo teisingumą (euristiniai algoritmai laikomi mažiau vertingais negu teisingi, bet neefektyvūs euristiniai), toks įvertinimas yra iš tiesų per mažas.

*50% taisyklė.* Atmetus buvusią 10–20% taisyklę, buvo pasirinkta kita schema, pagal kurią teisingos, bet neefektyvios programos turėtų surinkti ne mažiau kaip 50% galimų balų, o kiti 50% taškų būtų paskirstomi už efektyvumą. Taikant šį vertinimo principą buvo laikoma, kad *neefektyvios programos* vis dėlto yra *neefektyvios paisant protingų ribų* (pavyzdžiui, algoritmas, kuris rikiuoja seką tikrindamas kiekvieną galimą sekos narių perstatą, neišveiktų teisingumo tikrinimui skirtų testų). Šią schemą 2004 metais pasiūlė Tarptautinis mokslinis pasaulinių informatikos olimpiadų komitetas ir ji buvo taikoma dvejus metus – 2004 ir 2005 metais. Deja, ir ši schema sulaukė mokslininkų kritikos (Forišek, 2006; Opmanis, 2006). Jos pagrindiniai trūkumai tokie: 1) parašyti efektyvų sprendimą užima gerokai daugiau laiko negu neefektyvų; 2) parašę neefektyvų sprendimą dalyviai turi laiko konstruoti įvairias optimizacijas ir todėl dažniausiai surenka daugiau nei 50% taškų; 3) žinios ir įgūdžiai, reikalingi sukurti neefektyviam ir efektyviam sprendimui, smarkiai skiriasi, o skirtumas tarp gaunamų taškų už abu sprendimus yra per mažas. Atmetus ir šią taisyklę, 2006 metais pereita prie analogiškos *30–35% taisyklės*.

Kol IOI mokslinis komitetas keletą metų svarstė ir keitė įvairias vertinimo schemas, Kanados mokslininkai ėmėsi analizuoti IOI rezultatus taikydami statistinius tyrimo metodus (Cormac, 2006; Cormac *et al.*, 2006b; Kemkes *et al.*, 2006). Užuoat analizavę pačius uždavinius ar jų parinkimo veiksnius, jie nagrinėja užduotis kaip begalines panašių užduočių aibes tam tikra išraiška ir rezultatų analizei taiko pakopinį metodą, naudojami patikimumo intervalai, kurių hipotetinis išdėstymo teisingumas siekia 95%. Kanadiečių rezultatai parodė, kad nei užduočių sudėtingumas, nei procentinė išraiška, dėl kurios vyko diskusijos, neturi esminės įtakos daly-

\* Egzistuoja formalūs kriterijai, kuriais remiantis pasibaigus olimpiadai pagal dalyvių surinktus taškus įvertinama, kokio sunkumo buvo kiekvienas uždavinys: sunkus, vidutinio sunkumo ar lengvas.



vių tarpusavio išsidėstymui. Daug didesnę įtaką daro kokybiškas testų parinkimas (ypač akcentuojami testai, tikrinantys skirtingus sprendimo aspektus).

Kanados mokslininkų darbai atspindi vieną tyrimų kryptį, kuri siekia moksliniais metodais analizuoti ir remiantis tyrimų rezultatais automatinį testavimą padaryti objektyvesnį. Kita linkme dirbantys mokslininkai (Verhoeff, 2006) remiasi E. D. Dijkstros idėjomis, kad *testavimas gali būti naudojamas egzistuojančioms klaidoms aptikti, tačiau negali parodyti, kad jų nėra* (Dijkstra, 2005). Jie teigia, kad juodosios dėžės principu paremtas vertinimas negali būti objektyvus, ir siūlo nebijoti vėl įtraukti žmogiškąjį veiksnį į vertinimo schemas, ieškoti alternatyvių pirmiausia sprendimų pateikimo, o vėliau ir vertinimo būdų (Skūpienė, 2006; Skūpienė, Žilinskas, 2006).

## Apibendrinimas

Informatikos (algoritmų kūrimo) olimpiados rengiamos tarptautiniu mastu nuo 1989 metų. Jos sulaukia nemažai atgarsio mokslinėje visuomenėje ir tampa mokslinių tyrimų objektu. Ypač daug dėmesio skiriama automatinio testavimo juodosios dėžės principu problematikai. Per aštuoniolika olimpiados gyvavimo metų automatinis vertinimas informatikos olimpiadose perėjo tris etapus. Pirmuoju etapu (1989–2000) pereita nuo iš dalies automatizuoto prie visiškai automatizuoto vertinimo, nusistovėjo uždavinių formulavimo standartai, leidžiantys automatiš-

kai testuoti sprendimus. Deja, susiduriama su problema, kai dalyviams tenka daugiau dėmesio skirti sprendimui apipavidalinti negu pačiam algoritmui, nes darosi sunku užtikrinti, kad būtų tenkinami visi automatiniam testavimui būtini reikalavimai, ir nemažai programų (iš jų ir labai gerų) negauna taškų, jei apipavidalinimas neatitinka standartų. Šios problemos inicijavo Programavimo varžybų aptarnavimo sistemų atsiradimą (tai antrasis etapas, aprėpiantis 2001–2005 metus), kurios sudarė sąlygas patiems dalyviams per olimpiadą gana greitai pasitikrinti, ar jų programos apipavidalintos teisingai, ir leido vėl telkti dėmesį į olimpiados esmę – algoritmų kūrimą ir jų realizavimą. Trečiuoju etapu (nuo 2006 metų) imama suprasti, kad net ir atsiradus varžybų sistemoms automatinis vertinimas nėra visiškai objektyvus, pradedamos kurti ir taikyti įvairios vertinimo schemas, į programomis užrašytų algoritmų teisingumo ir efektyvumo vertinimo problematiką imama žvelgti moksliniais aspektais, atsiranda reikšmingesnių tyrimų ir publikacijų. Šiuo metu dirbama dviem kryptimis. Pirmosios krypties šalininkai yra tos nuomonės, kad automatinis testavimas yra vienintelis įmanomas vertinimo būdas informatikos olimpiadose, ir atlieka statistinius tyrimus, siekdami nustatyti jo patikimumą, pateikti tobulinimo rekomendacijas. Antrosios krypties atstovai mano, kad testavimas negali parodyti, jog klaidų nėra, ir atskleisti paties algoritmo, tad ieškoma alternatyvių algoritmo pateikimo ir vertinimo būdų.

## LITERATŪRA

CORMAC, G. (2006) Random factors in IOI 2005 test case scoring. *Informatics in Education*, vol. 5, no. 1, p. 5–14.

CORMAC, G.; KEMKES, G.; MUNRO, I.; VASIGA, T. (2006) Structure, scoring and purpose of computing competitions. *Informatics in Education*, vol. 5, no. 1, p. 15–36.

DAGIENĖ, V.; SKŪPIENĖ, J. (2004) Learning by competitions: Olympiads in Informatics as a tool for training high grade skills in programming. In *2nd International Conference on Information Technology: Research and Education*, London, 28 June–1 July, 2004. London Metropolitan University, 2004, p. 79–83.

DIJKSTRA, E.W. (2005) On the reliability of programs. Prieiga per internetą: <http://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD303.html> [žiūrėta 2007-05-17].

FORIŠEK, M. (2006) On the suitability of programming tasks for automated evaluation. *Informatics in Education*, vol. 5, no. 1, p. 63–76.

*Honk Kong Olympiads in Informatics*. Prieiga per internetą: [http://www.hkoi.org/training2006/ioitft\\_rules.php](http://www.hkoi.org/training2006/ioitft_rules.php) [žiūrėta 2007-05-17].

HORVATH, G.; VERHOEFF, T. (2002) Finding the median under IOI conditions. *Informatics in Education*, vol. 1, p. 73–92.

HORVATH, G.; VERHOEFF, T. (2003) Numerical difficulties in pre-university informatics education and competitions. *Informatics in Education*, vol. 2, no. 1, p. 21–38.

*International Olympiads in Informatics, official page*. Prieiga per internetą: <http://olympiads.win.tue.nl/oi/> [žiūrėta 2007-05-17].

*IOI 2002 Contest and Grading System*. Prieiga per internetą: <http://theory.snu.ac.kr/oi/> [žiūrėta 2007-05-17].

KEMKES, G.; VASIGA, T.; CORMAK, G. (2006) Objective Scoring for Computing Competition Tasks. In *Informatics Education – the Bridge between Using and Understanding Computers: Proceedings of International Conference in Informatics in Secondary Schools – Evolution and Perspectives*, November 7–11, Lecture Notes in Computer Science. Springer-Verlag, p. 230–241.

*Lietuvos moksleivių informatikos olimpiados, oficialus puslapis*. Prieiga per internetą: <http://ims.mii.lt/olimp/> [žiūrėta 2007-05-17].

OPMANIS, M. (2006) Some ways to improve olympiads in informatics. *Informatics in Education*, vol. 5, no. 1, p. 113–124.

PIELE, D. (2001) IOI'2001. Prieiga per internetą: <http://oldweb.uwp.edu/academic/mathematics/usaco/2001/oi/report.htm> [žiūrėta 2007-05-17].

*SIO.NET project*. Prieiga per internetą: <http://www.oi.edu.pl/sionet/> [žiūrėta 2007-05-17].

SKŪPIENĖ, J. (2006) Programming Style – Part of Grading Scheme in Informatics Olympiads: Lithuanian Experience. In *Informatics Education – the Bridge between Using and Understanding Computers: Proceedings of International Conference in Informatics in Secondary Schools – Informatikon Technologies at School*, November 7–11. Vilnius: TEV, p. 545–552.

SKŪPIENĖ, J.; ŽILINSKAS, A. (2006) Evaluation of programs in Informatics Contests: case of implementation of graph algorithms. In *Proceedings of The Third E-learning conference “Computer Science Education”*, Coimbra – Portugal, 7–8 September, 2006.

SKŪPIENĖ, J. (2004) Automatinis testavimas informatikos olimpiadose. Iš *Informacinės technologijos 2004: Konferencijos pranešimų medžiaga*. Kaunas: Technologija, p. 37–41.

VERHOEFF, T. (2006) The IOI is (not) a science olympiad. *Informatics in Education*, vol. 5, no. 1, p. 63–76.

## DEVELOPMENT AND PERSPECTIVES OF AUTOMATED GRADING IN INFORMATICS OLYMPIADS

### Jūratė Skūpienė

#### Summary

International Olympiads in Informatics (IOI) lately are gaining more attention in the scientific community. Contestants in IOI have to design algorithms and implement them as programs in one of allowed programming languages. Currently all the submissions are graded automatically using black-box method and grading is based on execution of compiled programs with different tests (input data). The algorithm itself is not analysed or revealed in any other possible way. Current grading system receives a lot of criticism due its unfairness (mistyping the name of a variable might lead to zero points), however, no better grading models have been proposed so far. The paper gives an overview of the development of grading in IOI starting from the very first IOI's where the verbal description of an algorithm had to be presented and evaluated. Many grading problems emerged in 1990's due to input/output format requirements which were inevitable in order to be able to per-

form automated testing of submissions. Input/output details required a lot of concentration from participants for mistyping a file name or redundant/missing *end of line* symbol might have resulted in zero points. Sometimes that even shifted focus from algorithm to formatting details. These problems were solved with the appearance of the first Informatics Contest Management System in 2001 which allowed the contestants to submit and run their program with sample tests thus checking their programs for compatibility with format requirements and correcting if there was such a need during contest time. After the contest management systems found their place in IOI's the attention was shifted to the relationship between the grades given for a submission and the algorithm implemented. The paper presents overview of published papers and discussions in IOI community on these topics. The paper ends with perspectives and directions for future improvement of grading in IOI's.